

Комплексне QA-тестування фронтенду — 3 доказами, по severity, на реальних девайсах

Ви шукаєте QA, який пройде сайт уздовж і впоперек: функціональність, UX, шрифти, кольори, адаптивність на всіх екранах, кросс-браузер, mobile і desktop. Робимо саме це — і не «на одному ноутбуку», а на матриці в'юпортів і трьох браузерних движках. Кожен дефект приходить до вас зі скріншотом, кроками відтворення й рівнем критичності. Готові стартувати ASAP. Нижче — як ми працюємо, а в кінці документа приклеєний витяг з нашого реального QA-звіту по practik.ua.

360 → 2560

В'ЮПОРТИ

5 ключових ширин: від малого смартфона до 2K-десктопа

3

БРАУЗЕРНІ ДВИЖКИ

Chromium, Firefox (Gecko), WebKit (движок Safari)

WCAG 2.2

ДОСТУПНІСТЬ

контраст кольорів, tap-targets, читабельність шрифтів

4

РІВНІ SEVERITY

● критично · ● високо · ● середньо · ● дрібно

ВАШ СКОУП — ЩО ПІД ЦИМ ПЕРЕВІРЯЄМО

Сім напрямів із вашого завдання, розкладені на конкретні перевірки

Спершу покажу, що завдання прочитане так, як ви його поставили. Зліва — ваші формулювання, справа — що саме під ними тестуємо на практиці. Скоуп уточнимо в особистих, але напрям роботи видно вже зараз.

ВАШЕ
ФОРМУЛЮВАННЯ

ЩО САМЕ ПЕРЕВІРЯЄМО

Функціональність

Усі сценарії: форми й валідація, пошук, фільтри, кошик і чекаут, кнопки, переходи, помилкові й порожні стани, граничні випадки (довгий текст, відсутні дані, повільна мережа). **Не «клікнули — відкрилось»**, а наскрізні шляхи користувача.

UX	Логіка й зрозумілість переходів, консистентність патернів між сторінками, стани <code>loading / empty / error / success</code> , мікроваємодії, зрозумілість для нового відвідувача. Фіксуємо, де користувач «застрягає».
Шрифти	Завантаження веб-шрифтів (ефект <code>FOUT/FOIT</code>), коректність fallback, накреслення й товщини, міжрядковий, читабельність на дрібних екранах, «з'їхані» гліфи й переноси.
Кольори	Контраст текст/фон за WCAG 2.2 AA , консистентність палітри, стани <code>hover / focus / active</code> , видимість фокуса, перевірка на типи дальтонізму.
Адаптивність на всіх девайсах	Прогін на 360 / 390 / 768 / 1920 / 2560 : коректні breakpoints, відсутність горизонтального скролу, обрізаного чи накладеного контенту, працездатність touch-жестів і орієнтацій.
Кросс-браузерне	Ті самі сценарії у Chromium, Firefox (Gecko) і WebKit (двигжок Safari) — ловимо розбіжності рендеру, поведінки форм, CSS-фіч, які «на одному браузері все добре».
Mobile + Desktop	Реальні в'юпорти, а не «звужене вікно»: touch проти hover, поведінка sticky-елементів і меню, клавіатура на мобільному, портрет/ландшафт.

МЕТОДОЛОГІЯ

Як ми тестуємо — п'ять кроків від карти сайту до звіту

Тестування — не «походив і поскаржився». Це повторюваний процес: спершу карта сценаріїв, далі систематичний прогін по в'юпортах і браузерах, потім глибокий шар, і на виході — структурований звіт, де кожен дефект підкріплений доказом.

КРОК 1

Recon і карта

Обходимо ключові екрани, будуємо карту сценаріїв і чек-лист під ваш скоуп.

КРОК 2

Multi-viewport

Прогін 5 в'юпортів x сторінки, скріншоти, фіксуємо зломи верстки й скролу.

КРОК 3

Кросс-браузер

Ті ж сценарії у 3 движках — ловимо розбіжності рендеру й поведінки.

КРОК 4

Глибокий шар

Функціональні сценарії, форми, WCAG-контраст і tap-targets, шрифти, симуляція швидкості.

КРОК 5

Звіт по severity







Кожен дефект: доказ-скріншот, кроки відтворення, рівень, пропозиція фіксу.

Матриця покриття: в'юпорти **360 · 390 · 768 · 1920 · 2560 px** × движки **Chromium · Firefox · WebKit** × ключові сценарії. Це дає те, що неможливо побачити на одному екрані — саме там і живуть «у мене все працює, а у клієнта зламано».

ЯК ВИГЛЯДАЄ КОЖЕН ДЕФЕКТ

Не «тут щось зламано», а структурований дефект із доказом

Звіт, з якого можна одразу працювати: кожна знахідка описана так, що і розробник розуміє «що чинити», і ви розумієте «чому це важливо». Один дефект — шість коротких блоків плюс рівень критичності й файл-доказ.

БЛОК ДЕФЕКТУ	ЩО В НЬОМУ
 Що бачить користувач	Симптом простими словами — що саме «не так» на екрані.
 Лабораторні дані	Конкретні числа: розмір елемента, контраст, час відгуку, в'юпорт і браузер, де відтворюється.
 Простими словами	Чому це проблема для звичайної людини, без технічного жаргону.
 Технічно	Першопричина для розробника: що в кодї/верстці/ресурсах це викликає.
 Пропозиція фіксу	Конкретний напрям виправлення, а не «треба полагодити».
 Бізнес-вплив	Як дефект б'є по конверсії, довірі чи доступності — щоб пріоритет був зрозумілий.

Severity + доказ обов'язкові. Кожен дефект має рівень **КРИТИЧНО** **ВИСОКО** **СЕРЕДНЬО** **ДРІБНО** і прикладений скріншот або відео з кроками відтворення. Жодного «здається, зламано» — тільки те, що відтворюється і доводиться. Внизу цього документа — приклад двох таких дефектів з нашого реального звіту.

ПРОЦЕС

Три фази — від карти тестів до звіту й ретесту

Фази йдуть послідовно: кожна спирається на готову попередню. Старт — ASAP, щойно отримаємо доступ і детальний скоуп. Обсяг і графік по фазах узгодимо після того, як звіriamo скоуп у особистих.

Фаза 1

Ресон і план тестів

- Обхід сайту, карта ключових екранів і сценаріїв
- Матриця в'юпортів і браузерів під ваш проєкт
- Чек-лист під ваш скоуп (функціонал, UX, шрифти, кольори, адаптив)
- Узгоджуємо пріоритетні зони й критерії приймання

Результат: прозорий план — що, де і як перевіряємо, до першого прогону.

Фаза 2

Прогін і фіксація

- Multi-viewport прогін (5 ширин) + скріншоти
- Кросс-браузер у 3 движках
- Функціональні сценарії, форми, граничні випадки
- WCAG-контраст і tap-targets, шрифти, симуляція швидкості
- Кожен дефект одразу з доказом і severity

Результат: повний реєстр дефектів із доказами — нічого «на словах».

Фаза 3

Звіт і передача

- Структурований звіт по severity (6-блочний формат)
- Priority-матриця: що чинити першим за впливом
- Розбір зі командою / розробником — за потреби
- Ретест після ваших фіксів — підтверджуємо, що закрито

Результат: ви знаєте точно, що зламано, наскільки критично і в якому порядку лагодити.

ЩО ВИ ОТРИМАЄТЕ НА РУКИ

Делверібли, з якими команда може працювати одразу

Звіт — не один pdf «для галочки». Це набір, де є і детальна картина для розробника, і пріоритети для прийняття рішень, і докази по кожному пункту.

Делверібл	Що всередині
✓ Інтерактивний звіт (HTML)	Усі дефекти з фільтрами по severity/розділу, докази, кроки відтворення
✓ PDF-звіт	Той самий звіт для друку й пересилання — зручно для узгоджень

Делверібл	Що всередині
✓ Priority-матриця (таблиця)	Severity x зусилля — наочно, що дає максимум при мінімумі правок
✓ Скріншоти-докази	По в'юпортах і браузерях, з підсвіченою проблемною зоною
✓ Ретест після фіксів	Перевірка закритих дефектів — підтвердження, що справді полагоджено

WEBCORELAB STUDIO

Ми це вже робимо — і робимо **доказово**

Studio у Торонто. З 2014 року будемо й тестуємо вебпродукти: фронтенд, верстка, інтеграції, QA-аудити з фізичними доказами. Свіжий приклад — повний QA-аудит **praktik.ua**: 25 знахідок, кожна зі скріншотом, кроками відтворення й рівнем критичності, на матриці в'юпортів і трьох браузерях. Витяг із цього звіту приклеєний у кінці документа.

2014

З цього року в розробці й QA

5000+

Реалізованих задач

Toronto

Команда Канада + США

QA-АУДИТ · FRONTEND

praktik.ua

25 дефектів із доказами, severity, multi-viewport + кросс-браузер — **приклад у кінці документа**

RESPONSIVE · CROSS-BROWSER

Адаптив і браузері

360 → 2560 px, Chromium/Firefox/WebKit — **знаходимо те, що видно лише в матриці**

UX · ACCESSIBILITY

UX і доступність

Шрифти, кольори, контраст, tap-targets за WCAG 2.2 — **повна картина фронтенду**

Чотири причини довірити QA саме нам



Доказова методологія

Кожен дефект — зі скріншотом і кроками відтворення. Не «здається зламано», а те, що відтворюється.



Реальна матриця

360 → 2560 px і три браузерні движки. Ловимо саме те, що не видно на одному екрані.



Severity і пріоритет

Сортуємо за впливом на бізнес — ви бачите, що чинити першим, а що може почекати.



Не тільки баги

UX, шрифти, кольори, доступність WCAG 2.2 — повна картина фронтенду, а не список збоїв.

Готові почати тестування ASAP

Цей документ — як ми працюємо й що ви отримуєте на руки. Скиньте детальний скоуп і доступ — складемо план тестів під ваш проєкт і стартуємо.

Питання й уточнення — відповім тут, у відгуку на проєкт. Наші роботи:

webcorelab.com/case-studies

— **Nebo**, WebCoreLab · webcorelab.com

ДОДАТОК · ПРИКЛАД НАШОЇ РОБОТИ

Витяг із QA-звіту **praktik.ua**

Нижче — фрагмент нашого реального QA-аудиту, щоб ви бачили формат, глибину й рівень доказовості. Повний звіт містив 25 знахідок із фізичним доказом по кожній. Тут показані два фронтенд-дефекти у нашому 6-блочному форматі та склад делверіблів. Бізнес-показники клієнта опущені з міркувань конфіденційності.

25

Знахідок із доказами

5

В'юпортів (360 → 2560)

3

Браузерні движки

5+

Делверіблів

РОЗПОДІЛ ЗА КРИТИЧНІСТЮ

25 знахідок, відсортованих за severity

Кожна знахідка отримала рівень критичності — щоб клієнт бачив не «купу проблем», а чіткий порядок: що чинити в першу чергу, а що може почекати.

9

● КРИТИЧНО

11

● ВИСОКО

4

● СЕРЕДНЬО

1

● ДРІБНО

Розділи: Технічне 14 · Мобільне 4 · Аналітика 3 · SEO 2 · CRO 1 · UX 1. Нижче — два показові фронтенд-дефекти повністю у нашому форматі.

F-001

Мобільна швидкість 38/100 — сторінка «думає» 2–4 секунди до реакції

КРИТИЧНО

🔧 ЩО БАЧИТЬ КОРИСТУВАЧ

На середньому смартфоні сторінка завантажилась візуально, але кілька секунд не реагує на тапи — людина встигає вирішити, що «висить».

📊 ЛАБОРАТОРНІ ДАНІ

Mobile Performance 38/100 (медіана 3 прогонів).
Total Blocking Time 2000–4000 ms.
Відтворюється на в'юпортах 360 і 390.

😬 ПРОСТИМИ СЛОВАМИ

Телефон «замерзає» на старті: картинка є, а натиснути ще не можна. Частина відвідувачів іде, не дочекавшись.

🔧 ТЕХНІЧНО

Важкий JS блокує головний потік на старті — звідси високий TBT і відкладена інтерактивність попри готовий перший екран.

✳️ ПРОПОЗИЦІЯ ФІКСУ

Розбити й відкласти некритичний JS (code-split, defer), прибрати важке з головного потоку, lazy-load для нижніх блоків, стиснути медіа першого екрана.

💰 БІЗНЕС-ВПЛИВ

Більшість трафіку — мобільна. Повільний старт прямо ріже залученість і конверсію саме там, де відвідувачів найбільше.

F-016

92% інтерактивних елементів <44px — пальцем важко попасти (WCAG 2.2)

ВИСОКО

🔧 ЩО БАЧИТЬ КОРИСТУВАЧ

На сторінці заявок кнопки й посилання стоять тісно — палець влучає в сусідній елемент замість потрібного.

📊 ЛАБОРАТОРНІ ДАНІ

92% інтерактивних елементів менші за 44×44 px (вимога WCAG 2.2 SC 2.5.8 та Apple HIG).
В'юпорти 360/390, відтворюється стабільно.

😬 ПРОСТИМИ СЛОВАМИ

Цілі для пальця замалі: тиснеш — потрапляєш не туди. Дратує і змушує перетискати по кілька разів.

🔧 ТЕХНІЧНО

Недостатній hit-area й spacing інтерактивних елементів на мобільному; падіння WCAG 2.2 SC 2.5.8 (Target Size).

✳️ ПРОПОЗИЦІЯ ФІКСУ

Збільшити зону натискання до 44×44 px (padding/min-height), додати відступи між сусідніми діями, не зменшуючи візуальну щільність.

💰 БІЗНЕС-ВПЛИВ

Це сторінка генерації лідів — кожен промах по кнопці = ризик втратити заявку на найважливішому кроці.

Делверібли по practik.ua: інтерактивний HTML-звіт із усіма 25 дефектами · PDF-версія для друку · priority-матриця (severity × зусилля) · окрема технічна вибірка · карта шляху користувача (CJM) · слайди для презентації команді. Кожен дефект — з reproducible-кроками й скріншотом-доказом.

WebCoreLab · Toronto · QA Frontend Testing · приклад: practik.ua · 2026